



IBM Research

Math Libraries for BlueGene/L

Ramendra K. Sahoo
Blue Gene systems software
IBM Research

Feb. 24, 2005

© 2005 IBM
Corporation

Outline

- List of Math libraries.
- What the users should keep in mind while choosing math functions
- Benefits from Compiler and how to proceed
- Math library single node performance (Don't take it seriously!)
- Future Plans

List of Math Libraries

- Engineering & Scientific Subroutine Libraries (ESSL)
 - ❖ Only Static libraries
 - ❖ No shared libraries
- Mathematics Accelerated Scientific Subroutines (MASS)
- Mathematics Accelerated Scientific Subroutines Vectorized (MASSV)
- FFTW

ESSL for BG/L

- Based on ESSL 4.2 for p-series Linux/AIX.
- Uses same code/ algorithms used for other ESSL releases.
- Core routines optimized to exploit higher order compiler optimizations (-O4, -O5).
- Optimized to provide maximum benefit of 440d (double hummer)
- Always use `-qarch=440d, -qtune=440` to get the double hummer code generations..(not necessarily always would provide performance benefits!).

ESSL Modules

■ Linear Algebra Subprograms	(I)	(S)	(L)
❖ Vector-scalar	0	41	41
❖ Sparse vector-scalar	0	11	11
❖ Matrix-vector	1	32	32
❖ Sparse matrix-vector	0	0	3
■ Matrix Operations			
❖ Addition, subtraction, multiplications, rank-k updates, rank-2k updates and transpose	0	25	26
■ Linear Algebra Equations			
❖ Dense linear algebraic equations	3	53	58
❖ Banded linear algebraic equations	0	18	18
❖ Sparse linear algebraic equations	0	0	11
❖ Linear least squares	0	3	5

ESSL Modules (Contd..)

■ Eigensystem Analysis

- ❖ Solutions to general eigensystems & general eigensystem analysis problems

(I)	(S)	(L)
0	8	8

■ Signal Processing Computations

- ❖ Fourier transforms
- ❖ Convolutions and correlations
- ❖ Related Computations

0	15	11
0	10	2
0	6	6

■ Sorting and Searching

- ❖ sorting, sorting with index, & binary and sequential searching

5	5	5
---	---	---

ESSL Modules (Contd...)

	(I)	(S)	(L)
■ Interpolation <ul style="list-style-type: none">❖ Polynomial and cubic spline interpolation	0	4	4
■ Numerical Quadrature <ul style="list-style-type: none">❖ Numeric quadrature on a set of points or on a functions	0	6	6
■ Random Number Generation <ul style="list-style-type: none">❖ Generating vectors of uniformly distributed random numbers	0	3	3
■ Utilities	8	0	3
■ Total	13	240	253

Planning Your Program

- Select an ESSL subroutine
- Avoid Conflicts with Internal ESSL Routine Names Exported
- Setting up your data
- Setting up your ESSL calling sequences
- Using auxiliary storage in ESSL
- Providing a correct transform length in ESSL
- Getting the best accuracy
- Getting the best performance
- Dealing with errors while using ESSL

Planning Your Program

- An ESSL subroutine is a named sequence of instructions within the ESSL product library whose execution is invoked by a call.
- Interpreting the subroutine names with a prefix underscore

Example :

`_GEMUL` (all versions of the matrix multiplication subroutine
`SGEMUL`, `DGEMUL`, `CGEMUL` and `ZGEMUL`)

S for short-precision real, D for long-precision real

C for short-precision complex Z for long-precision complex

I for integer

- Syntax

fortran	<code>CALL NAME-1 NAME-2 ... NAME-n (arg-1, arg-2,..., arg-m,...)</code>
C & C++	<code>name-1 name-2 name-n (arg-1, ..., arg-m,...);</code>

Planning Your Program (Contd..)

- Conflicts with Internal ESSL routines :
Avoid using **ESV** as prefix names for your names.
- Scalar data passed to ESSL from all types of programs, including Fortran, C, and C++, should conform to **ANSI/IEEE 32-bit** floating point format as per ANSI/IEEE standard for binary floating-point arithmetic, ANSI/IEEE Standard.
- All arrays, regardless of the type of data, should be aligned to ensure optimal performance. **Alignment exceptions** can be figured out through compilation options.

Planning Your Program (Contd..)

- Conflicts with Internal ESSL routines :
Avoid using **ESV** as prefix names for your names.
- Scalar data passed to ESSL from all types of programs, including Fortran, C, and C++, should conform to **ANSI/IEEE 32-bit** floating point format as per ANSI/IEEE standard for binary floating-point arithmetic, ANSI/IEEE Standard.
- All arrays, regardless of the type of data, should be aligned to ensure optimal performance. **Alignment exceptions** can be figured out through compilation options.

ESSL Functional Testing

- We have functional tests carried out with a number of different options
 - ❖ -O3 440 (98% success)*.
 - ❖ -O3 440d (95% success)*.
 - ❖ -O5 440d (91% success)*.

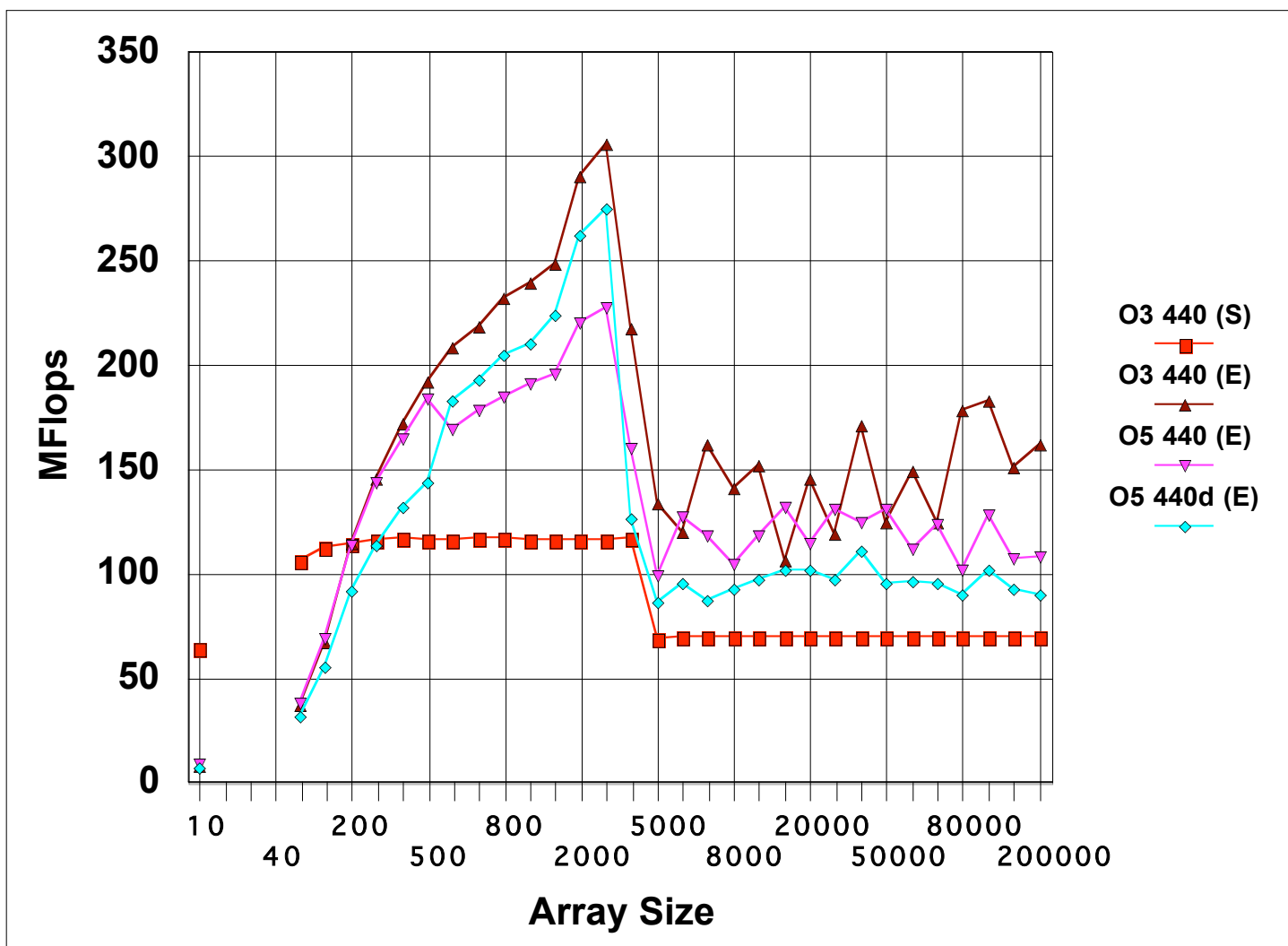
- A number of outstanding defects fixed in compilers (particularly the TPO, TOBEY related).

* Based on results for ESSL 4.1 in 2004.

Example Routine (DASUM)

- `SUM = DASUM (N,DX,INCX)`
- Compute the sum of the absolute values in the vector.
- A comparison of results from vanilla code with ESSL.
- Example codes : `dasum.F` (removing `qstrict`, O3)
 `dasum_vanilla.F` (code from netlib)
 `dasum_orig.F` (code from ESSL)
- Use `-qdebug=diagnostic` to examine which loops are simdized.
- **Limitation** : among loops with strides, simdization only possible for stride 1 loops.
 Hence : Vanilla `dasum` code performance is better than ESSL !
- **Solution** : Adding Pragmas to take care of non stride 1 loops (available in future compiler releases)

ESSL sample performance results



MASS/MASSV Libraries

- Provides elementary math functions in both scalar and vector form
 - ❖ Examples : sqrt, pow, inv, log etc.
- Provides trigonometric and hyperbolic math functions in scalar and vector form.
- Examples : sin, cos, tan, atan, sinh

- All the routines are C routines (replacing Assembly routines written for p-series).
- A list of functions already supported :
 - ❖ Scalar functions : atan, exp, rsqrt, tanh, sincos, cosh, log, sinh, sqrt, pow, tan
 - ❖ Vector functions : vacos, vcos, vlog1p, vsasin, vsexpm1, vslog, vssinh, vasin, vlog, vsatan2, vsexp, vspow, vssin, vatan2, vdiv, vpow, vscbrt, vsincos, vsqrt, vssqrt, vcbt, vrcbrt, vscosh, vsinh, vsrbrt, vstanh, vcosh, vexpm1, vrec, vscosisin, vsin, vsrec, vstan, vexp, vrsqrt, vscos, vslog10, vsrsqrt, vtanh, vcosisin, vlog10, vsacos, vsdiv, vslog1p, vssincos, vtan

Sample MASS library performance benefits

Function	Libmass.a (cycles)	Libm.a (cycles)	
sqrt	42.0	101.0	
rsqrt	35.0	133.0	
exp	56.0	168.0	
log	68.0	316.7	
sin	66.6	191.9	
cos	65.8	199.9	
tan	89.5	316.5	
atan	109.0	216.0	
sinh	81.0	326.1	
cosh	68.0	239.4	
pow	157.0	521.3	

Summary

- We provide the same set of math libraries for BG/L as provided in other IBM platforms .
- Functionality part has been tested and verified.(2004)
- Math libraries will have significant performance improvements in next few months.(Specifically ESSL release for BG/L targetted for October 2005)
- Higher optimizations (-O4/ -O5) a reality !
- A number of compiler fixes and improvements (including special pragmas supporting math functions).
- Need your feed back in terms of performance/tuning results to further improve.